A REAL-TIME TELEMETRY SIMULATOR OF THE IUS SPACECRAFT

Michael E. Drews, Douglas A. Forman, Damon M. Baker
Octant Technologies, Inc.

IN-17
376504

Louis B. Khazoyan
Boeing Information, Space and Defense Systems

Danilo Viazzo
Integrated Systems, Inc.

ABSTRACT

A real-time telemetry simulator of the IUS spacecraft has recently entered operation to train Flight Control Teams for the launch of the AXAF telescope from the Shuttle. The simulator has proven to be a successful higher fidelity implementation of its predecessor, while affirming the rapid development methodology used in its design. Although composed of COTS hardware and software, the system simulates the full breadth of the mission: Launch, Pre-Deployment-Checkout, Burn Sequence, and AXAF/IUS separation. Realism is increased through patching the system into the operations facility to simulate IUS telemetry, Shuttle telemetry, and the Tracking Station link (commands and status message).

KEY WORDS

Telemetry Simulators, Commercial Off The Shelf (COTS) Systems, Real-Time Simulation, Inertial Upper Stage (IUS), and Space Shuttle

INTRODUCTION

The Inertial Upper Stage (IUS) is an unmanned, two-stage high altitude booster used in conjunction with either the Shuttle or Titan launch vehicles. It is propelled by two solid rocket motors, inertially navigated, and controlled in three axes. The IUS, pictured in Figure 1 below, can deliver a spacecraft of over 5000 pounds from the Shuttle or Titan IV Park orbit to geostationary orbit. The IUS has also been used to launch spacecraft on interplanetary trajectories. The IUS includes significant redundancy in its systems, to improve reliability. There are two versions of the IUS avionics currently in the inventory. The first, which will fly on the STS-93/IUS-27/AXAF (Advanced X-Ray Astrophysics Facility) mission, uses dual computers and a separate Redundant Inertial Measurement

Unit (RIMU). The second, which will fly on the TIVB/IUS-21/DSP mission, uses an integrated Flight Controller (FC) with MIL-STD 1750A processors and ring-laser gyros.

The IUS program utilizes a sizable Flight Control Team (FCT) made up of government and contractor personnel. In order to assure that this team is prepared to respond to in-flight anomalies, a number of launch simulations are performed prior to each mission. These rehearsals involve the use of a simulator to generate nominal IUS telemetry and allow the introduction of simulated failures. The FCT identifies the failure, generates plans for mitigating the problem, and executes the plans, including transmission of commands to the simulated vehicle to modify the vehicle configuration.

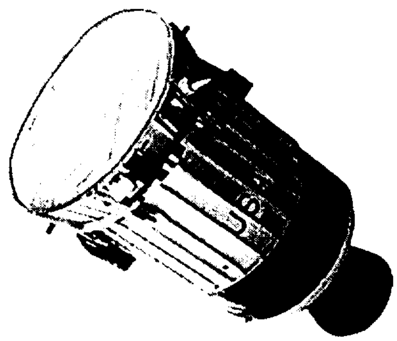

Figure 1: The IUS Spacecraft

In December 1997, the original database driven system used to simulate IUS telemetry was upgraded to a much higher fidelity dynamic simulator. The realism of launch exercises using the simulator is enhanced by the fact that the unit is patched into the commanding and telemetry systems of the operations facility, Onizuka Air Station. Across this "patched presence", the simulator produces three separate telemetry streams: IUS spacecraft, Shuttle Customer Ancillary Services (CAS), and the bi-directional Automated Remote Tracking Station (ARTS) link. IUS telemetry is periodic, frame synchronized, with sub-sub commutation, contains 1200-1600 measurands, and has selectable rates of 16 or 64 KBPS. The Shuttle CAS frame is a packetized, NRZL stream, populated with 9 NASCOM blocks, and is sent at a rate of 56 KBPS, The simulated ARTS link provides an interface for vehicle commands, and broadcasts a 1/sec status message at 9.6 KBPS.

There are two versions of the new IUS Telemetry Simulation System (I-TSS), one for the Flight Controller/Titan configuration (IFTSS), and one for the RIMU/Shuttle/AXAF configuration (IATSS). Both systems were constructed using the same design methodology, which involves a unique integration of Commercial Off The Shelf (COTS) hardware and software, yet simulates the complete mission scenario. In the case of the IATSS, this includes Launch, Pre-Deployment-Checkout, Deployment from the Shuttle, a series of SRM, RCS and Vernier burns, and AXAF/IUS separation. To date, the system has completed several simulation exercises, ranging from six to ten hours in duration. This paper outlines not only the performance capabilities of the operational system, but also how these capabilities were rapidly and cost effectively implemented utilizing commercially available software and hardware.

Development Methodology

The design of the IUS telemetry simulator involved the integration of COTS hardware components in with software developed around the use of Computer Aided Software Engineering (CASE) tools. The Matrix_X SystemBuild graphical programming environment and AutoCode code generation tool were used to develop approximately 75% of the coded subsystems. The graphic environment also provided a platform to synthesize code developed not only in a block diagram format, but hand-written code and legacy code. For the IUS simulator, the calculation of data to mimic the spacecraft and space environment was coded in the graphical environment. However, a database method was utilized to actually process this data into a telemetry format, which was more convenient and efficient to hand code. The hand-coded subsystems are represented as "code blocks" at the graphic level, which enables the graphic environment to act as the focal point for software integration. Once converted from FORTRAN to C, portions of the IUS flight code were similarly integrated as "code blocks" at the graphic level. Developing code in this fashion allowed specific sub-systems to be constructed in the most efficient manner possible, while gathering them in an easily documented and intuitively navigable environment.

This methodology relies heavily on the rapid prototyping capability provided by automatic code generation tools. In general, designs can be modified and quickly tested when re-coding a specific implementation involves only a simple command line directive. Additional features of code generators include the ability to use the same graphic code to produce code in multiple languages and for multiple operating systems as well. For the IUS telemetry simulator, the graphical portion of the code generated 77,407 lines of C code, which was compiled and linked with other individually compiled "code blocks" (23,368 lines), forming the executable code and scheduler.

Regarding hardware, the IUS simulation system was designed such that it could make use of exclusively commercially available components. This focussed the development process on integration issues, reducing cost and time associated with designing low volume, application specific hardware. An industry standard VME backplane was chosen for the simulator chassis. The basis of this decision stems from the wide ranging availability of products and customer support as much as the specific performance capabilities of VME systems. Similarly, PowerPC based VME boards, and Greenspring IP-MP products were used to accomplish the bulk of the processing and I/0 functions. The system architecture is shown in Figure 2 below, and detailed in the following section.

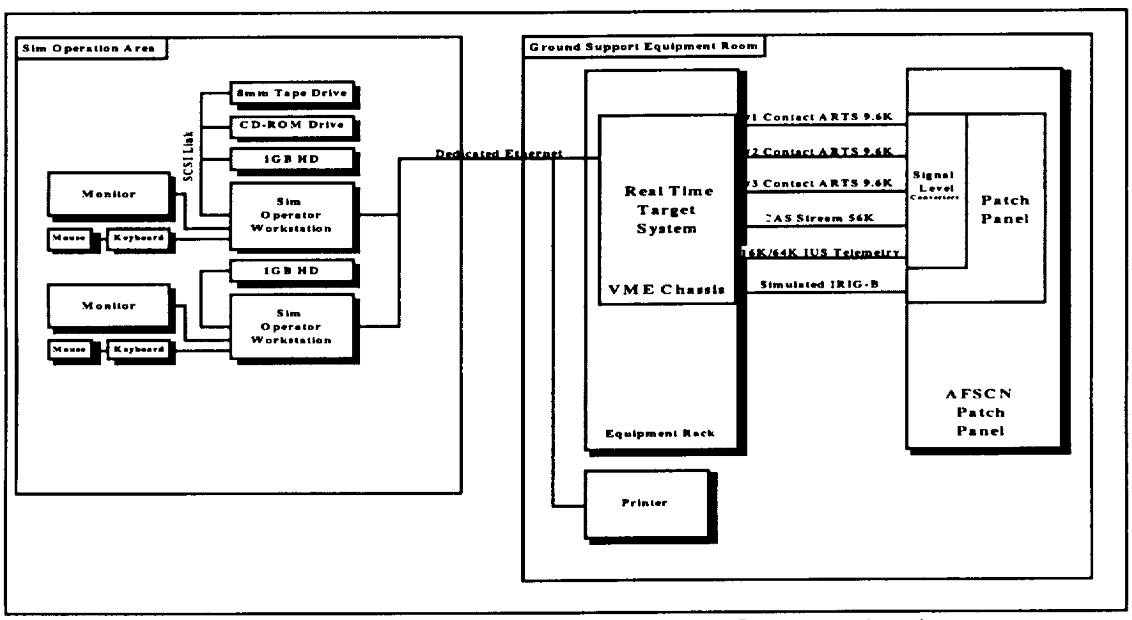

Figure 2: Simulator Architecture and Communications

System Architecture

The IUS simulator architecture has two primary nodes. The "host" system, typically a workstation, is where code development takes place. The "target" system is a multi-processing platform where the simulation code runs in real-time. The host system also functions as the GUI interface to the target, so that an operator may interact with the simulation once it is initiated. An ethernet connection is used to connect the target and host system, and several protocols are used to complete the code transfer. The executable is sent across via TFTP, and the telemetry and simulation initialization databases are transferred by NFS. Once the code is downloaded, the simulation connects the Graphical User Interface (GUI) on the host to the target across a TCP/IP net, termed "ianet", which is also used by the Scriptor/text based interface.

The target system supports three basic functions: real-time processing of the code, communication with the host and I/O with the operations facility. The target system utilizes a VME back plane, and three PowerPC 604e 100 MHz processors. The executable code is downloaded to the three processors through transition modules, which provide the interface from the host/ethernet to the target/VME environment. One VME serial carrier board houses four IP-MP serial HDLC cards, one for each of the telemetry streams (IUS, Shuttle and ARTS) and one for command reception. An IRIG-B board allows the simulator GMT time to be synched with that of the facility.

Although there are two simulator configurations, the hardware architecture between them remains exactly the same, and the bulk of the code retains similar functionality. The majority of the simulation code can be broken up into five categories:

A. Models that describe the space environment and vehicle dynamics
B. Models that describe the degree to which the IUS hardware can measure and interact with the space environment
C. Models of the flight software and flight CPU, that demonstrate how the software interprets information from the hardware, and the ground, and then emulates the appropriate decision making algorithms
D. Models and code that assemble and transmit spacecraft telemetry
E. Models and code that simulate how the spacecraft receives and executes commands originating from the ground segment, astronauts or flight software

Dynamics and Spacecraft Models

Category A models, also termed "Truth" models, are constructed from first principal equations. They keep track of the space environment as well as the dynamics governing the vehicles and other modeled components of the simulation. Model examples include:

- Six Degree of Freedom (6DOF) Dynamics
- Quaternion propagation of attitude, including the Euler Angle equivalents
- Orbit position, computed using J5 gravity perturbation equations for an oblate Earth, and solved for using a fourth order Runge-Kutta method
- Summation of forces and moments due to solar and lunar position, as well as vehicle Solid Rocket Motor (SRM) and Reaction Control Propulsion (RCS)
- Changes in the Center of Gravity due to spacecraft mass changing as a function of accumulated thruster on-times and staging

The spacecraft hardware models, which make up Category B, represent approximately 35% of the telemetry stream, and include elements of the attitude control and determination system, as well as the overall propulsion systems and instrument packages onboard the spacecraft. The IUS has five accelerometers and five gyros, which are modeled to the degree that they take into A/D, D/A conversion of delta-angles and delta-velocities to counts. There are 12 RCS thrusters, used for attitude control and Vernier burns, and two SRM motors. Many of the systems have the capability to allow anomalies to be induced "on-the-fly", such as failing the upstream or downstream portion of the valve on any of the 12 thrusters, on either the A or B side of the spacecraft.

A large number of the hardware telemetry points for IUS are associated with the Electrical Power System (EPS) of the vehicle. The EPS was modeled in great detail to capture the behavior of the major electrical components. A generic battery model was used to simulate the battery output voltage versus the current charge. Graphic control is provided to the simulator operator to control the efficiency and the charge left for each battery. The battery model is used to drive the bus model. The simulator includes models for Stage 1 and 2 Avionics Bus, Stage 2 Utility Bus, Stage 1 Spacecraft Bus, and an Ordnance Bus for both the primary and redundant sides of the IUS. Bus current is determined by the sum of the current generated by the fixed and switched loads of each bus. A Power Distribution Model is used to propagate the effects of commands to hardware devices. Telemetry measurements and scaling are performed to prepare the data to be used by Signal Conditioning Unit (SCU) model, which passes data to the telemetry collection code.

The SCU model includes the signal conditioning models for data that is passed to telemetry, and also the processing required to properly identify and execute commands received from either the command link or the flight software. The SCU model also is duplicated to simulate the primary and redundant configurations found in the vehicle's hardware. The telemetry signal conditioning function of the SCU includes models for analog and digital muxes. Each mux can be failed so that all the associated channels are set to a desired state by using the simulator GUI. Bus voltage is also used to scale analog signals and resolve the state of digital signals. Commands are executed via a Relay Matrix. The state of the individual Relay Matrix channels on either primary or secondary sides can be forced to a specific state through the simulator GUI display, allowing the operator the capability to simulate the failure of specific commands. This causes the FCT to operate the remainder of the mission with a reduced level of commanding.

In the IATSS version, there are additional models to simulate the EPS interface between the orbiter and IUS, and the 6DOF Dynamics relating the Shuttle and IUS while mated in orbit. The GUI interface to the simulator contains a series of panels that accurately replicate the functionality of the Aft Flight Deck panels in the Orbiter. These include the Power Control Panel (PCP), the Communication Interface Uplink (CIU), Payload Retention Control (PRC) and Standard Switch Panels 1 and 2 (SSP1, SSP2). The bulk of the interaction involves shadowing Astronaut interaction as they manipulate the PCP to power up the IUS/AXAF, and transfer it to internal power. Through the CIU, the "Sim Astronauts" can also send commands to the IUS, and through using the PCP, they can raise the IUS/AXAF on the Tilt Table to a 58 degree inclination, release the umbilicals, and fire the pyro devices, all in preparation for deploy. The "Sim Pilot" can maneuver the Orbiter to a deploy attitude, or any other orientation that is definable as a pitch, yaw, and roll offset from either a Local Vertical, Local Horizontal (LVLH) or Inertially Held (IH) reference frame. The Shuttle models interact with telemetry in two ways. First, state vector, attitude and similar information regarding the Orbiter fills the appropriate portions of the Shuttle CAS telemetry stream. Second, the IUS Ground Support Equipment (GSE) compares in real-time both IUS and CAS telemetry.

IUS Flight Software

Type C Models simulate the functionality of the IUS flight software, which includes guidance, navigation, control, mission sequencing, communications and on-board testing. In some circumstances, it was deemed a less complicated task to re-host IUS software from available FORTRAN or Ada code, instead of simulating the flight software functionality with graphically generated code. For example, Gamma Guidance, which does the on-board targeting for the IUS, is a non-linear equation solver, which requires approximately 10,000 lines of FORTRAN. In the simulator development process, a bulk FORTRAN-to-C conversion was accomplished, followed by hand review and verification of the functionality. This was significantly easier than attempting to model the complex formulation of the Gamma Guidance algorithms at the graphical programming level.

Many other flight software functions are not nearly so complex, and were emulated using the graphical programming environment. These include attitude control, mission sequencing, and navigation. Attitude control includes different control modes for Thrust Vector Control (TVC), used during SRM burns, and the RCS thrusters used at other times. Mission sequencing involves several linked lists of mission events, which are executed in the appropriate sequence to activate various IUS functions. Navigation includes mapping from 5 sensors to three-body axis, allowing introduction of low-level sensor anomalies. An additional function added for the IATSS is the Pre-deploy checkout. This is an automatic series of tests performed while still in the Shuttle bay that verifies IUS readiness for flight. These tests exercise the TVC and Relay Matrix on both of the redundant avionics strings. A combination of graphical and hand-written code is used to mimic the checkout function and allow the insertion of a number of pre-built anomalies.

Telemetry Simulation

The states of flight software values are assembled in conjunction with data from the spacecraft hardware models to populate the telemetry frames. Models and hand code that construct the telemetry frames are termed Type D models. Figure 3 depicts the assemblage of NASA and Air Force components that the IATSS simulates during a launch rehearsal.

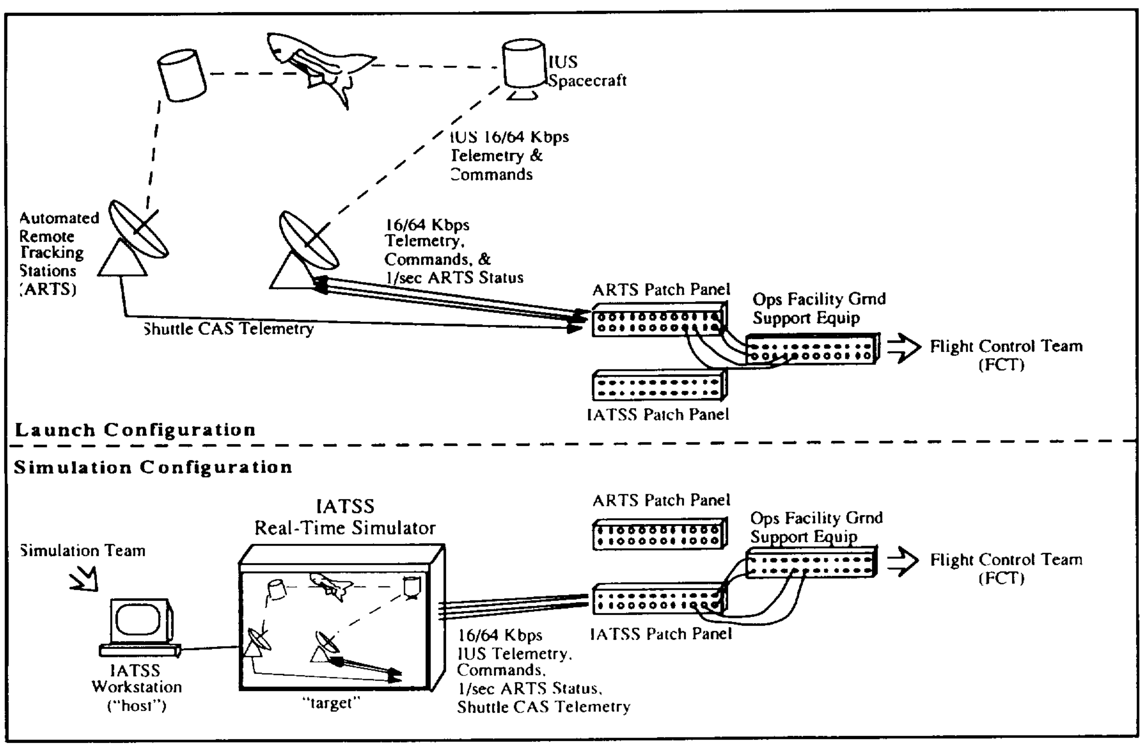

Figure 3. "Patched Presence" of IATSS at Operations Facility

For the IUS vehicle, the simulator produces two telemetry formats, 64K and 16K. Both of these formats are continuously synchronous, periodic, sub-sub commutated, NRZL coded bit streams. The 64K wave train consists of 4 major frames. Each 64K major frame consists of 50 minor frames. Each 64K minor frame consists of 160 eight-bit words. The entire 64K master frame contains 32000 8-bit words, and takes 4 seconds to complete. A given measurand in the 64K format will report in telemetry at a rate of 50, 25, 2, 1, .5, or .25 Hertz. The 16K wave train consists of 8 major frames. Each 16K major frame consists of 25 minor frames. Each 16K minor frame consists of 80 eight-bit words. The entire 16K master frame contains 16000 eight-bit words and takes 2 seconds to complete. A given measurand in the 16K format will report in telemetry at a rate of 25, 13, 12, 1, .5, .25, or .125 Hertz. The simulator collects and outputs measurands from the models so that measurands of the same rate groups, especially the navigation measurands, are consistent, synchronized sets, even though they may appear in different minor and major frames.

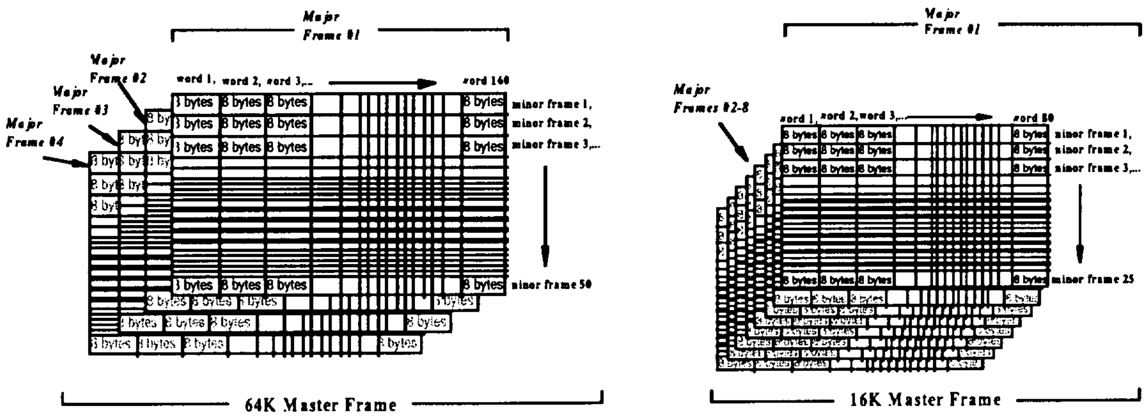

Figure 4. Description of IUS Telemetry Frames

The telemetry stream is configured by reading flat ASCII human readable text files during initialization, and constructing run-time in-memory databases optimized for use during real-time execution. Three types of database files are used to configure the simulated IUS telemetry stream:

- For each telemetry format there is a wave train file, 64K.wtr and 16K.wtr. These files specify the frame sizes, synch patterns, and frame counter parameters.
- The User Data Base (UDB) file describes all telemetry measurands. For IUS, a perl script was written to automatically build the UDB file from information in the IUS Master Telemetry File (MTF). This file contains an entry for each measurand, stipulating its unique simulation index, bit length, start bit and data type. This entry also contains wave-train parameters for each wave-train format, including word offsets, and intervals within the minor, major and master frames. Aliases for measurands are also specified in this file.
- The SystemBuild User (SBU) files are used to interface between the graphic code environment and the hand coded telemetry output code. Each of these files corresponds to a specific hand coded "code block" in the graphic code environment that collects a subset of modeled telemetry points. An SBU file contains a list of Simulation Indexes corresponding to the list of signal names (which are telemetry measurands or aliases) used in the graphic code environment.

A telemetry conversion function is installed on each measurand instance based on the measurand's data type (read in from the UDB file). This conversion function is responsible for converting measurand engineering values produced by the graphic code environment, to the binary form that the actual IUS vehicle outputs in telemetry (based on the data type of the measurand). The following data types are used in the IUS telemetry simulator: 0) Unsigned Integer (Arbitrary length) 1) Delco Floating Point 32 bit Single Precision 2) Delco Floating Point 64 bit Double Precision 3) 1750 Floating Point 32 bit Single Precision 4) 1750 Floating Point 48 bit Double Precision 5) Two's Complement.

The DOD IFTSS and "clear" IATSS simulators both have the same telemetry wave-train formats, so they both use the same 16K.wtr and 64K.wtr files. However, each of these IUS vehicles has its own MTF file, with different measurand locations and data types. Therefore a different simulator UDB file is generated for each mission. One major difference is that AXAF has a Delco flight computer, so it outputs Delco Floats in telemetry, and FC has a 1750A computer, so it outputs 1750 Floats.

The IATSS also supports a CAS (Customer Ancillary Services) telemetry stream for Shuttle Orbiter data. The CAS data is transmitted at a rate of 56K bits per second and uses NRZL encoding. One CAS message is transmitted per second. Each CAS message is formatted into NASCOM blocks. The number of NASCOM blocks contained in the CAS messages can range from one to eleven, and is determined by the number of blocks needed to contain the several measurands in the CAS message. The IATSS is currently configured for 9 NASCOM blocks.

The ARTS stations, in addition to receiving telemetry from the vehicles, produce a telemetry stream or status message, once they have been signaled by the operations facility. Contact is made by sending a Set Asynchronous Response Mode (SARM) message, and the ARTS station responds with a 1/sec status message. The contents of this message include antenna pointing information, receiver lock, uplink power, range and rate information, and the status of the link configuration (command tones, frequency, modulation indexes etc). In addition to the Type D code that assembles the status message, there is Type A (dynamics) and Type B (hardware) models at work, simulating the site visibility, closed loop antenna tracking of the IUS, atmospheric signal attenuation, and antenna gain patterns. There are three ARTS stations model, which can be run simultaneously, allowing for "hot hand-overs". The position/identity of the ARTS can be switched on the fly to any of eleven pre-programmed locations, or set manually with latitude, longitude and altitude GUI panel inputs.

Spacecraft Commanding

A portion of the ARTS code can also be categorized as Type E (Command Modeling) because the ARTS stations are the primary command path used during a launch rehearsal. The Shuttle CIU panel mentioned above is also used to send commands to the IUS. Two additional command methods are available, which allow the simulation team to inject commands at any time: through a GUI Commanding Panel, and through the Scriptor, which provides an alternative text based interface to the entire simulation environment. When commands are received through the ARTS path, they are transmitted from the operations facility Ground Support Equipment (GSE), and read into the IATSS by a serial card. A driver decodes the serial stream and allows the command software to match up the commands with the IUS Command Database, and trigger the appropriate functionality, such as changes in software values, or enabling the EPS/SCU to generate voltages to fire off Relay Matrix commands. Commanding from the simulator GUI panel interface is as easy as entering a four digit command ID on that panel that matches the IUS On Orbit Handbook (OOH) ID for that command. The command is injected into the models of the IUS command handling software, and the Vehicle Command Count (VCC) incremented, just as if the command had come across in the conventional manner.

The Scriptor allows commands to be entered into the IUS model in the same manner as the Command Panel, except that instead of a windows/GUI type of interface, a text based/command line interface is provided. The functionality of the Scriptor, however, goes far beyond an alternative IUS commanding interface. The Scriptor provides the user with a mechanism for overall simulation control, script file invocation and the capability to adjust the values in the simulation models. Simulation control consists of pausing, resuming, Check-pointing, data collection, and fast-forwarding the simulator. Script file invocation consists of entering the name of a text file "script" that will be parsed and executed at a particular Mission Elapsed Time (MET). Directives within these scripts usually alter the value of a model or telemetry parameter, and can be executed immediately, or set to occur at a time tag referenced in the script. The altered points can reflect either steady state values, or a dynamic response, called a "trend."

Trending model values consists of using highly parameterized mathematical functions (including sinusoidal, polynomial, and exponential) to either drive simulation inputs, or alter the behavior of nominal telemetry. ALL IUS or CAS telemetry measurands can be trended by the Scriptor, even if they are not modeled. If a trend is applied to a modeled telemetry point, the modeled value is over-ridden until the trend duration is expired. The Scriptor syntax allows the user to trend telemetry values using the actual IUS OOH textual mnemonic, or any alias for that point that exists in the simulation UDB file. For example, the user can trend telemetry points which may be less than 8 bits, and starting anywhere in a telemetry word, without having to be aware of what the value of that telemetry word is, or what other telemetry points may affect that word. Another example of Scriptor trend usage is a telemetry point may be trended to have an appropriate pre-launch behavior until launch, at which point the "flight" behavior modeled in the simulator takes over.

Launch Simulation Operations

IUS flight operations consist of a pre-deploy phase prior to separation from the booster and a flight phase. In a Shuttle mission, the pre-deploy phase may last more than seven hours, while for a Titan, the flight phase begins approximately nine minutes after launch. The IUS Flight Control Team includes personnel from Boeing, Lockheed Martin, Air Force Space Command, and Air Force Materiel Command or NASA. In addition, the IUS FCT communicates with other centers, including the spacecraft control center, the launch base, and with Johnson Space Center (JSC) for Shuttle operations.

The IUS simulator is used during all FCT exercises in preparation for a launch. There are several different levels of exercises that are used to build the team's readiness for launch. The first exercises/launch simulations are internal, IUS only, "standalone" exercises. After several of these exercises, the spacecraft center is included in several "inter-center" rehearsals. Finally, for Shuttle operations, all centers, including JSC, participate in several Joint Integrated Simulations that include astronaut crewmembers "flying" the Shuttle simulator in Houston.

Conclusion

As of April 1998, the IATSS and IFTSS have completed ten successful launch simulations, each lasting six to ten hours. With each simulation, the Flight Control Team provides feedback that allows the simulation developers using the rapid prototyping methodology to continually enhance the fidelity and accuracy of the simulations. New anomalies are also provided for the simulation team, and work continues to enhance the overall system. The primary improvements now ongoing include a Checkpointing feature, and faster than real-time simulation. Check-pointing will provide users with the ability to save the thousands of states within the simulator to a file, allowing the simulation team to pre-run simulations, and begin launch exercises at any Mission Elapsed Time of their choice. The simulation operators will also be able to modify anomalies repeatedly before an exercise, without waiting up to 8 hours each time to propagate the mission to the same place as previously. Faster than real-time simulation will provide similar benefits and allow simulation exercises to be shortened if a portion of the exercise is deemed un-educational enough to warrant "fast forwarding."

Implementation of fast forwarding represents the utility of using COTS equipment. In this case, common PC processors hosted on VME cards are continually undergoing industry funded performance enhancements. The consistency or backwards compatibility of interfaces in commercial products, and cost reduction based on high volume sales, are two factors that allow upgrading with minimized risk and financial impact. Development of the IUS simulator has also served as a forerunner to telemetry simulation and Hardware In The Loop systems currently being developed for the Air Force Phillips Laboratory, and China Lake Naval Air Warfare Center.

Acknowledgments

The authors would like to acknowledge the efforts of the engineers at OTI, ISI and Boeing that developed or contributed to the IFTSS and IATSS simulators, as well as the IUS Simulation team at Onizuka Air Station.

References

Brandenburg, James, "External Interface Control Document JSC-GSFC Operational Communication ICD for Mission Control Center systems", JSC-11534 Vol. I, Rev C, United Space Alliance, Houston, Texas, April, 1998.

Hardtla, John W., "Gamma Guidance for the Inertial Upper Stage," American Institute Aeronautics and Astronautics Guidance and Control Conference, Palo Alto, California, August 1978

Wertz, James R. Spacecraft Attitude Determination and Control. D. Reidel Publishing Company, Inc. Hingham, MA, 1995, pp. 510-579.